One View, Many Worlds: Single-Image to 3D Object Meets Generative Domain Randomization for One-Shot 6D Pose Estimation

Zheng Geng[*,1], **Nan Wang**[1], **Shaocong Xu**[1],
Chongjie Ye[1,5], **Bohan Li**[6,7], **Zhaoxi Chen**[4], **Sida Peng**[2], **Hao Zhao**[†,1,3]

[1] Beijing Academy of Artificial Intelligence, BAAI; [2]Zhejiang University
[3] Institute for AI Industry Research (AIR), Tsinghua University
[4] Nanyang Technological University; [5]FNii, The Chinese University of Hongkong, Shenzhen
[6]Shanghai Jiao Tong University; [7]Eastern Institute of Technology, Ningbo

zhenggeng@bit.edu.cn* zhaohao@air.tsinghua.edu.cn†

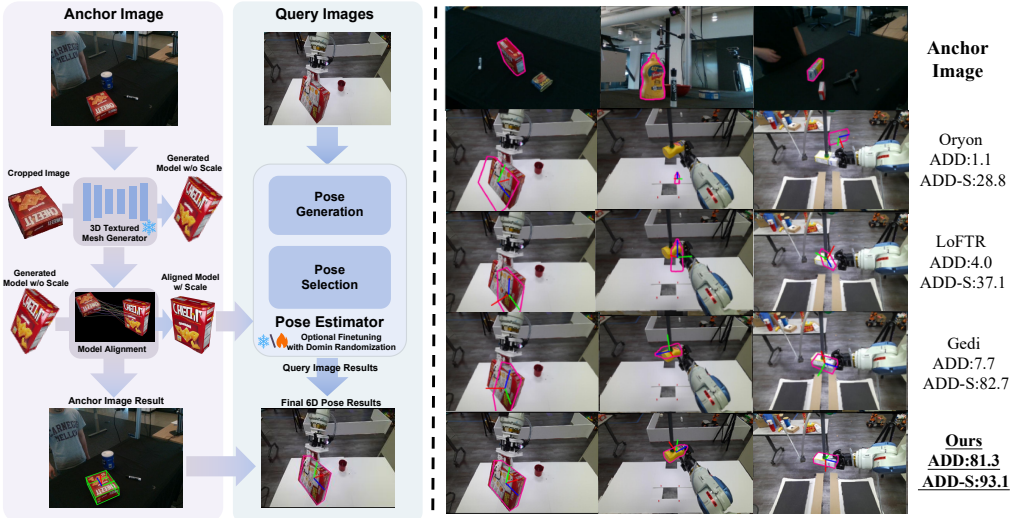

Figure 1: **OnePoseViaGen.** (Left) From a single anchor image, we generate a textured 3D model that lacks real-world scale and pose. Our coarse-to-fine alignment resolves this challenge, and optional domain-randomized fine-tuning further boosts robustness. (Right) While prior methods (Oryon, LoFTR, Gedi) largely fail in this one-shot setting, our approach achieves dramatic gains (ADD 81.3, ADD-S 93.1), enabling reliable 6D pose estimation where baselines break down.

Abstract: Estimating the 6D pose of arbitrary unseen objects from a single reference image is critical for robotics operating in the long-tail of real-world instances. However, this setting is notoriously challenging: 3D models are rarely available, single-view reconstructions lack metric scale, and domain gaps between generated models and real-world images undermine robustness. We propose OnePoseViaGen, a pipeline that tackles these challenges through two key components. **First**, a coarse-to-fine alignment module jointly refines scale and pose by combining multi-view feature matching with render-and-compare refinement. **Second**, a text-guided generative domain randomization strategy diversifies textures, enabling effective fine-tuning of pose estimators with synthetic data. Together, these steps allow high-fidelity single-view 3D generation to support reliable one-shot 6D pose estimation. On challenging benchmarks (YCBInEOAT, Toyota-Light, LM-O), OnePoseViaGen achieves state-of-the-art performance far surpassing prior approaches. We further demonstrate robust dexterous grasping

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

with a real robot hand, validating the practicality of our method in real-world manipulation. Project page: https://gzwsama.github.io/OnePoseviaGen.github.io/

Keywords: Object Pose Estimation, Generative Model, Robot Manipulation

1 Introduction

Robots operating in the open world must interact with an endless long-tail of novel objects—ranging from an unseen tool on a factory line to an ad-hoc household item in daily life. At the core of this capability lies 6D object pose estimation, the recovery of an object's precise 3D position t and orientation R. However, this task remains notoriously difficult in realistic settings: pre-scanned CAD models are rarely available, multi-view capture is impractical, and single-view reconstructions suffer from scale ambiguity. Consequently, reliable one-shot 6D pose estimation from a single image has long been deemed nearly impossible, despite its central role in simulation [1–3], tracking [4], reconstruction [5–19], and downstream robotic applications such as manipulation [20–45], navigation [46, 47], autonomous control [48, 49] and teleoperation [50, 51]. While VLA models and generalist policies [52–61] excel in broad skills such as folding or bussing, they often fail in precision-critical operations—for example, inserting a charging plug—where sub-centimeter errors lead to task failure. Accurate 6D pose estimation is thus indispensable for reliable robotics.

Most state-of-the-art 6D pose estimation methods are learning-based and can be classified into three categories: instance-level [62–65], category-level [66–71], and category-agnostic [72–75]. While instance-level methods achieve high accuracy, they are restricted to objects with rich texture in the training set, limiting their use in long tail applications. Category-level methods generalize better within known categories but still struggle with unseen object classes. Recent advances in category-agnostic methods focus on pose estimation for unseen objects using render&compare strategies [74, 76], feature matching [73, 77], or both [72], achieving high precision. However, these methods depend on 3D models, which are not trivial to acquire in most scenarios.

To tackle the lack of object models, recent work in model-free pose estimation explores methods that avoid reliance on explicit textured 3D models. These approaches use reference images and have shown promising results, with some employing partial matching for relative pose estimation [78–80] or multiple images to reconstruct the object [74, 81]. However, many still require multi-view inputs or prior camera-to-object pose knowledge [72, 80], limiting their effectiveness when only a single image is available but robust 6D pose estimation is needed.

To address these challenges, we propose OnePoseViaGen, a novel model-free method for 6D object pose estimation. Building on single-image 3D generation techniques [82–84], we extend Hi3DGen [82] with image-conditioned generation to reconstruct textured 3D object models from a single RGB-D image. Since such reconstructions are scale-normalized, we introduce a coarse-to-fine strategy that jointly refines object scale and 6D pose for accurate alignment. By integrating the refined model with advanced pose estimation frameworks [74], our method achieves robustness under heavy occlusion, complex lighting, and large viewpoint variations. We evaluate OnePoseViaGen on standard benchmarks (YCBInEOAT, Toyota-Light, LM-O) and a new annotated in-the-wild dataset, and further demonstrate its effectiveness in dexterous grasping tasks. Results show that our method significantly outperforms state-of-the-art approaches for unseen objects. Our contributions are summarized as follows:

- **Generative pipeline for one-shot 6D pose.** We introduce the first pipeline that integrates single-view 3D generation into both training and inference for one-shot 6D pose and scale estimation, proving that generative modeling can directly benefit pose estimation.
- **Coarse-to-fine metric alignment.** We design a coarse-to-fine alignment module that jointly refines pose and metric scale via 2D–3D feature matching and render-and-compare refinement, enabling accurate metric-scale alignment from a single image.

- **Text-guided generative domain randomization.** We propose a text-driven augmentation strategy that generates diverse, structurally consistent 3D variants. Rendering these models under randomized conditions yields a large-scale synthetic dataset, which bridges the domain gap between generated models and real-world objects.
- **State-of-the-art performance and real-world validation.** Our method establishes new state-of-the-art results on three challenging 6D pose benchmarks and a newly collected in-the-wild dataset. We further demonstrate its effectiveness in dexterous robotic grasping, validating robustness and practicality in real-world manipulation.

2 Related Works

2.1 6D Pose Estimation

The field of 6D pose estimation has been extensively explored through both model-based and model-free approaches, each addressing the challenge of localizing objects in 3D space with varying strategies [72–74, 76, 85–89]. Model-based methods typically rely on CAD models or 3D representations for pose estimation. For instance, MegaPose [76] employs a render&compare strategy to refine poses of novel objects using synthetic views of CAD models. Similarly, FoundPose [73] leverages foundation features [90–92] from self-supervised vision models to establish 2D-3D correspondences between RGB images and pre-rendered templates, demonstrating robust generalization without task-specific training. FS6D [75] extends this domain by addressing few-shot pose estimation, introducing a dense prototype matching framework and a large-scale photorealistic dataset (ShapeNet6D) to enhance generalization. In contrast, model-free approaches aim to reduce reliance on explicit 3D models. Wang et al. [86]designed loss functions for specific symmetry types. OnePose [88] and its successor, OnePose++ [93], eliminate the need for CAD models by reconstructing sparse or semi-dense object models from RGB videos, enabling real-time pose estimation even for low-textured objects. GigaPose [72] achieves fast and accurate pose estimation by leveraging discriminative templates and patch correspondences, offering significant speedups while maintaining robustness to segmentation errors. FoundationPose [74] further advances the field by proposing a unified framework that supports both model-based and model-free setups, utilizing neural implicit representations for novel view synthesis when CAD models are unavailable. Gen6D [87] introduces a generalizable model-free approach that predicts poses of unseen objects using only posed RGB images. Despite these advancements, existing methods are limited to textured objects and require costly retraining for novel instances. In this work, we propose a novel model-free approach that leverages textured 3D model generation to achieve accurate pose estimation without prior 3D data.

2.2 3D Model Generation

The generation of high-fidelity 3D models from 2D inputs has emerged as a critical area of research [82, 94–102], with significant implications for applications such as 6D pose estimation. Hi3DGen [82] introduces a novel framework that leverages normal maps as an intermediate representation to bridge the gap between 2D images and detailed 3D geometry. By decoupling low- and high-frequency image patterns through noise injection and dual-stream training, Hi3DGen achieves superior fidelity in reproducing fine-grained geometric details, outperforming existing methods in generating high-quality 3D models. Similarly, TRELLIS [95] presents a scalable approach to 3D generation by integrating sparse 3D grids with dense multiview visual features, enabling versatile decoding into formats like Radiance Fields, 3D Gaussians, and meshes. Rapid 3D Model Generation highlights the potential of AR/VR devices as intuitive tools for 3D modeling, introducing Deep3DVRSketch [96], which enables novice users to create detailed 3D models from simple sketches efficiently. VP3D [97] enhances text-to-3D generation by incorporating 2D visual prompts, improving visual fidelity and texture detail through explicit visual appearance guidance. IT3D [98] further refines this process by synthesizing multi-view images explicitly, tackling issues like over-saturation and inadequate detailing via a Diffusion-GAN dual training strategy. Cycle3D [99] presents a unified framework that cyclically utilizes 2D diffusion and 3D reconstruction modules to

ensure high-quality and consistent 3D content generation. StyleTex [100] focuses on style-guided texture generation for 3D models, employing a diffusion-model-based approach to harmonize textures with reference images and text descriptions. Consistent3D [101] addresses inconsistencies in text-to-3D generation by exploring deterministic sampling priors through ODE trajectory sampling, ensuring more reliable and high-fidelity outputs. However, these methods generate normalized models with scale mismatches to real-world objects, leading to errors in downstream pose estimation. Our OnePoseViaGen introduces a coarse-to-fine strategy that jointly refines the scale and 6D pose of generated models, enabling accurate recovery of real-world object dimensions and scene poses.

3 Method

3.1 Problem Formulation

As illustrated in Fig. 2, let $\mathbf{I}_A$ (top-left corner) and $\mathbf{I}_Q$ (top-right) denote an RGB-D anchor image and a query image, respectively. Our task is to estimate the relative rigid transformation $\mathbf{T}_{A\to Q} \in SE(3)$ that maps the pose of a target object from the anchor image coordinate system to that of the query image. Since generated models are scale-normalized, we additionally determine a scaling factor s based on $\mathbf{I}_A$ to calibrate the model to real-world scale. Formally, $\mathbf{T}_{A\to Q} = [\mathbf{R} \mid \mathbf{t}]$. The core challenge lies in achieving robust pose estimation when the query image may capture the same object under significantly different conditions. Our method thus needs to extract invariant geometric features and establish reliable correspondences despite such appearance changes.

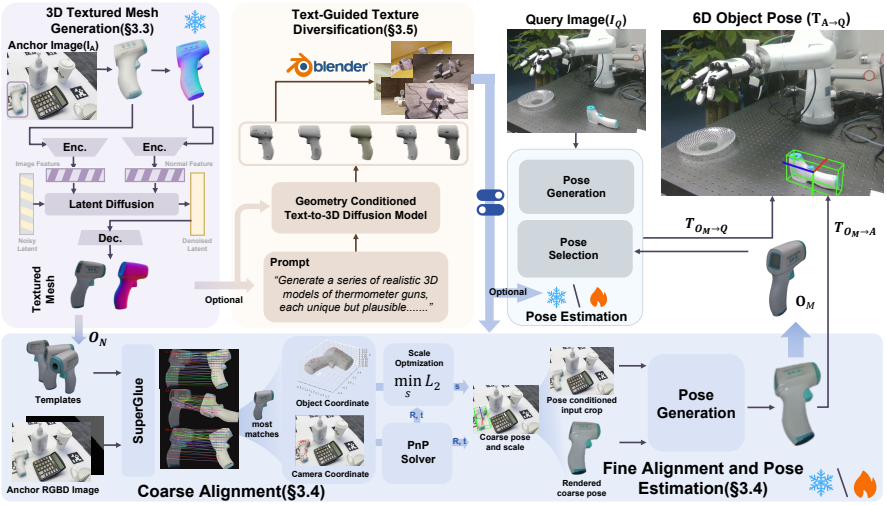

Figure 2: **Overview of OnePoseViaGen.** Given an anchor RGB-D image $\mathbf{I}_A$, we generate a textured 3D model O_N via our normal-guided generative pipeline. To ground O_N in real-world metrics, we align it with $\mathbf{I}_A$ via multi-view feature matching and render&compare framework. For a query image $\mathbf{I}_Q$, we apply a render&compare&selection pipeline [74]. Finally, the relative transformation $\mathbf{T}_{A\to Q}$ is computed. To enhance generalization, we introduce a text-prompt-based diversification strategy for training data, bridging the domain gap between generated models and real objects.

3.2 Approach Overview

Fig. 2 outlines our OnePoseViaGen pipeline. Given an RGB-D anchor image $\mathbf{I}_A$ (top-left of Fig. 2) of a novel object—for which no pre-existing 3D model exists—our key challenge is to enable 6D pose estimation from a single view. To address this, we first generate a textured 3D model with standardized orientation/scale from $\mathbf{I}_A$ using Hi3DGen[82] in single-view 3D generation (Sec. 3.3). Critical to grounding this model in the real world, our coarse-to-fine alignment module (Sec. 3.4) jointly recovers the object's metric scale and 6D pose in $\mathbf{I}_A$. With this metric-calibrated model, we estimate the object's pose in any query RGB-D image $\mathbf{I}_Q$ (top-right of Fig. 2) using the aligned model and a robust pose estimation framework. The final relative transformation $\mathbf{T}_{A\to Q}$ is then computed from the absolute poses in both views.

To bridge the domain gap between generated models and real-world images, we introduce a text-guided generative augmentation strategy (Sec. 3.5). This generates structurally consistent but texture-diverse 3D variants, which are rendered under randomized conditions (lighting, backgrounds, occlusions) to create a large synthetic dataset. This dataset enables fine-tuning of pose estimation components, significantly boosting robustness—validated in our experiments (Sec. 4).

3.3 Normal-guided 3D Textured Mesh Generation

Motivation. Estimating the 6D pose of novel objects typically requires a 3D model, which is often unavailable. While multi-view reconstruction can generate models from multiple images with known poses [74, 75, 87, 93], this is not feasible in our one-shot setting. Neural implicit representations can also generate 3D geometry but often require time-consuming optimization for each object [103–105]. To overcome these limitations and enable pose estimation from a single view, our first step is to generate a textured 3D object model directly from the input anchor image I_A. Leveraging recent advances in single-view generative 3D modeling allows us to quickly obtain a plausible 3D representation without the need for multi-view data or lengthy per-object optimization.

Standardized Textured Mesh Geneartion with Normal and Image conditions. As depicted in the "3D Textured Mesh Generation" step in Fig. 2, we generate high-fidelity textured meshes from a single image based on a modified version of Hi3DGen [82]. Initially, segmentation is used to crop the input anchor image to minimize background noise's impact on the model generation process. Subsequently, the processed image I_A^{cropped} undergoes an off-the-shelf image-to-normal estimation to produce a normal map X. The normal maps X and the original cropped image I_A^{cropped} are then fed into the 3D generation model. This process ensures geometric consistency and visual detail.

Finally, we obtain a standardized textured object model O_N in an object-centric coordinate system. This model O_N serves as the basis for subsequent alignment step to match real-world dimensions, as outlined in Sec. 3.4. We further visualize several examples of these generated models in Appendix. D, demonstrating the fidelity and diversity achieved.

3.4 Coarse to Fine Object Alignment and Pose Estimation

Motivation. Although Sec. 3.3 provides a 3D shape O_N, it is in a normalized space, not matching the real object's scale and pose in $\mathbf{I}_A$. For accurate metric-space 6D pose estimation, determining the correct scale and aligning the model with the observed object in the 3D scene is essential. This alignment is challenging due to discrepancies between the idealized model and noisy, partial observations from $\mathbf{I}_A$. To address this, we propose a multi-stage coarse-to-fine alignment strategy, illustrated in Fig. 3.

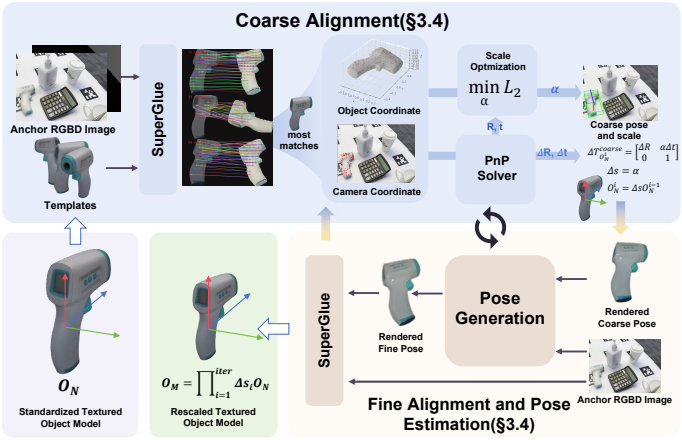

Figure 3: **Overview of Coarse-to-fine Alignment Process.**

Coarse Alignment via Multi-view Feature Matching and PnP solving. In coarse stage, we render object O_N from n spherical viewpoints using camera intrinsics $\mathbf{K}$. SuperPoint [106] features are extracted and matched via SuperGlue [107] between each rendered view o^i and $\mathbf{I}_A$, and the view with the most matches is selected. Matched 2D point pairs $(p'_i \in \mathbb{R}^2, p_i \in \mathbb{R}^2)$ (from the rendered

view and $\mathbf{I}_A$, respectively) are lifted to 3D points ($\mathbf{P'_i} \in \mathbb{R}^3, \mathbf{P_i} \in \mathbb{R}^3$) in the object's normalized coordinate system and the camera coordinate system using depth information from the rendered view and $\mathbf{I}_A$: $\mathbf{P'}_i^N = \frac{1}{d'_i} \cdot \mathbf{K}^{-1} \cdot [\mathbf{p'}_i^N, 1]^\top, \quad \mathbf{P}_i^N = \frac{1}{d_i} \cdot \mathbf{K}^{-1} \cdot [\mathbf{p}_i^N, 1]^\top$. Subsequently, we apply the Perspective-n-Point (**PnP**) algorithm [108] on $\mathbf{P'_i}$ and their corresponding 2D projections $\mathbf{p}_i$ in image o^i to estimate an initial 6-DoF pose, with scale ambiguity.

Then, we transform normalized model point $\mathbf{P'_i}$ into the same camera coordinate system: $\hat{\mathbf{P}}_\mathbf{i} = R\mathbf{P'_i} + t$. Ideally, $\hat{\mathbf{P}}_\mathbf{i}$ and corresponding camera-space points $\mathbf{P}_i$ represent the same 3D point, differing only by a global scale factor $\alpha > 0$, where α denotes the scaling factor from the origin of the camera coordinate system along the viewing direction: $\mathbf{P}_i^N = \alpha\hat{\mathbf{P}}_\mathbf{i}$.

To estimate α, we minimize the following least-squares objective: $L_2 = \sum_i \left\| \alpha\hat{\mathbf{P}}_\mathbf{i} - \mathbf{P}_i^N \right\|^2$. This leads to a single-variable optimization problem: $\min_\alpha L_2$, which yields the optimal α. And due to the scale consistency property of similarity transformations, the final model's scale remains invariant to the choice of scaling center. This α also serves as the desired scale factor s, defined with respect to the model center as the origin. Based on this optimization, we obtain a coarse estimate of the object's pose $\mathbf{T}^{\text{coarse}}_{O_N \to A} = \begin{bmatrix} R & \alpha t \\ \mathbf{0} & 1 \end{bmatrix}$. and scaling factor $s = \alpha$. Since the process is iterative, we also express the transformation incrementally: $\Delta\mathbf{T}^{\text{coarse}}_{O_N \to A} = \begin{bmatrix} \Delta R & \alpha\Delta t \\ \mathbf{0} & 1 \end{bmatrix}, \quad \Delta s = \alpha$.

Fine Alignment and Pose Estimation via Render-and-Compare Refinement. The coarse alignment provides a reasonable initial pose and scale, but inaccuracies due to feature noise and model imperfections necessitate further refinement. We iteratively refine the pose and scale using a strategy that combines render&compare refinement with the coarse alignment's scale optimization. Starting with the coarse estimate $\mathbf{T}^{\text{coarse}}_{O_N \to A}$ and scale s, we refine the estimation using the network adapted from FoundationPose [74] to predict incremental pose updates $\Delta R \in SO(3)$ and Δt based on the difference between a rendering of the model at the current pose and the observed object in $\mathbf{I}_A$. The pose is updated as $t^+ = t + \Delta t, \;\; R^+ = \Delta R \otimes R$, where $\otimes$ denotes update on SO(3). Crucially, after each pose update, we re-run the feature matching and scale optimization from the coarse stage using the current pose estimation to refine the scale s. This iterative process, alternating between pose refinement and scale optimization, continues until convergence or a maximum number of iterations is reached. This yields a metric-scale 3D model $O_M = \prod_i^{iter} \Delta s_i O_N$ and its accurate 6D pose $\mathbf{T}_{O_M \to A}$ relative to the anchor image frame $\mathbf{I}_A$. The effectiveness of this coarse-to-fine approach, particularly the fine alignment stage, is demonstrated in the ablation study (Sec. 4.4).

For estimating the poses in query images I_Q (illustrated in Fig. 2), we utilize the derived metric-scale model O_M. We employ a render&compare&selection strategy from FoundationPose [74]. The resulting 6D pose relative to the camera frame in query image I_Q is denoted as $\mathbf{T}_{O_M \to Q}$. Finally, as shown in Fig. 2, the relative transformation between the anchor and query views, is computed using the estimated absolute poses:

$$\mathbf{T}_{A \to Q} = (\mathbf{T}_{O_M \to A})^{-1} \cdot \mathbf{T}_{O_M \to Q}. \tag{1}$$

The quantitative results showcasing the accuracy of our 6D pose estimation are presented in Table 1, Table 2, and Table 3 in Sec. 4.

3.5 Text-Guided Texture Diversification

Motivation. The 3D model O_N generated in Sec. 3.3 captures shape and initial texture, but represents only a single instance. It lacks variation in appearance and geometry found in real-world objects, limiting its generalization under diverse environmental conditions (e.g., lighting, occlusion). Training robust pose estimators requires large, diverse datasets, which are costly to collect manually. To bridge the gap between generated models and real-world objects and facilitate effective training, we propose generating a diverse set of plausible object variations.

Texture Diversification with Structure-Aware Text-to-3D. As illustrated in the "Generative Domain Randomization" step in Fig. 2, we use the text-guided 3D generation model Trellis [84], taking the initial model O_N and text prompts as input (Appendix. A). The model generates variants with diverse textures, styles, or minor geometric changes while preserving core structure. We render these models under randomized viewpoints, lighting and occlusions using Blender, forming a large-scale synthetic dataset for fine-tuning the render-and-compare network (Sec. 3.4). The method improves the pose generation model on the long-tail category through diverse texture synthesis. Incorporating category-level training data enables the pose estimation module to significantly improve its performance—particularly on objects that were previously challenging, or when the occluded side appears in query images. As a result, the overall generalization capability of the pose estimation module is notably enhanced. As shown in Table 4, this significantly improves pose estimation performance.

4 Experimental Results

Table 1: Comparison with SOTA on the YCBInEOAT Dataset.

	Oryon [79]		LoFTR [78]		Gedi [109]		Any6D [110]		Ours	
Modality	RGB-D & Language		RGB-D		Depth		RGB-D		RGB-D	
Metrics	ADD-S(↑)	ADD(↑)	ADD-S(↑)	ADD(↑)	ADD-S(↑)	ADD(↑)	ADD-S(↑)	ADD(↑)	ADD-S(↑)	ADD(↑)
sugar_box1	44.0	0.0	47.3	0.0	95.6	0.0	96.7	14.3	93.71	75.63
sugar_box_yalehand0	34.7	3.0	41.6	0.0	82.2	6.9	89.1	75.2	94.59	87.12
mustard0	48.6	0.0	47.3	20.3	100	0.0	100	23	96.98	70.00
mustard_easy_00_02	36.2	0.0	23.2	0.0	78.3	0.0	78.3	53.6	85.58	70.12
bleach0	10.4	0.0	55.2	0.0	74.6	0.0	98.5	68.7	95.84	92.41
bleach hard_00_03_chaitanya	24.4	6.7	60	15.6	66.7	62.2	73.3	51.1	94.39	91.35
tomato_soup_can_yalehand0	32.8	0.0	10.7	0.0	60.3	0.0	70.2	0.0	88.77	77.72
cracker_box_reorient	13.2	0.0	26.3	0.0	97.4	0.0	100	60.5	96.34	79.40
cracker_box_yalehand0	15.0	0.0	22.6	0.0	89.5	0.0	97.7	63.9	91.64	87.64
MEAN	28.8	1.1	37.1	4.0	82.7	7.7	89.3	45.6	**93.10**	**81.27**

Table 2: Comparison on the TOYL Dataset.

Methods	AR(↑)	MSSD(↑)	MSPD(↑)	VSD(↑)
SIFT[111]	30.3	39.6	44.1	7.3
Obj. Mat. [112]	9.8	13.0	14.0	2.4
Oryon [79]	34.1	42.9	45.5	13.9
Any6D [110]	43.3	55.8	58.4	15.8
Ours	**55.7**	**67.0**	**65.1**	**35.1**

Table 3: Comparison on the LINEMOD Occlusion (LM-O) Dataset.

Methods	AR(↑)	MSSD(↑)	MSPD(↑)	VSD(↑)
GigaPose [72]	17.5	35.8	9.0	7.6
Any6D [110]	28.6	36.1	32.0	17.6
Ours	**74.8**	**79.5**	**84.9**	**60.0**

4.1 Datasets and Setup

Public datasets. We evaluated our method without the proposed fine-tuning (Sec. 3.5) to ensure fairness on three public datasets: YCBInEOAT [113] (robotic interaction), Toyota-Light (TOYL) [114] (challenging lighting), and LINEMOD Occlusion (LM-O) [115] (occluded, textureless).

Real-world evaluation. We performed two experiments in real-world settings: (1) 6D pose estimation for uncommon objects by generating synthetic training data via our domain randomization pipeline and testing on a calibrated real set (built based on [114]), and (2) robotic manipulation tasks, establishing grasping setups using a ROKAE robot arm equipped with an XHAND1 dexterous hand, and two AgileX PiPERs, and measuring success rates against baselines.

4.2 Comparison with the State of the Art

Quantitative results. Following the evaluation settings of Any6D [110], we align the relative pose estimate $\mathbf{T}_{A\to Q}$ according to Eq. (1). Our approach is primarily compared against recent state-of-the-art methods, including Any6D [110], Gedi [109], LoFTR [78], and Oryon [79]. On YCBInEOAT, anchor images are sampled from DexYCB [116]. As shown in Table 1, our method achieves significantly better robustness. Notably, on challenging objects such as {sugar_box1, mustard0, tomato_soup_can_yalehand0}, previous methods achieved very low ADD scores (e.g., Any6D: 14.3%, 23%, 0.0%), while our method consistently performs well (75.63%, 70.00%, 77.72%), achieving a mean ADD of 81.27. For TOYL (Table 2) and LM-O (Table 3), we report BOP benchmark metrics. Our method shows consistent improvements across all evaluation criteria.

4.3 Experimenting in Robotic Manipulation

To validate the practical applicability and robustness of our method, we conducted two real-world robotic manipulation tasks using 6D pose estimates as the perception input (See Appendix.D for details). (1) Pick-and-place: A dexterous hand mounted on a robot arm performs grasping. This task is challenging due to heavy occlusions during grasp. (2) Dual-arm manipulation: (pick, handoff, place): This involves coordinated control, object instability during transfer, and dynamic occlusions. We tested both tasks on 15 objects over 30 trials each, comparing our method with SRT3D [94] and DeepAC [117]. As shown in Fig. 4, our method achieved a success rate of 73.3% (Table 5), demonstrating strong performance in complex, real-world scenarios.

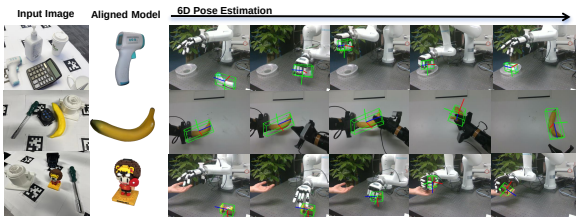

Figure 4: **Qualitative results of robot manipulation.** The left column is the anchor images, the middle column shows the generated models, and the right part shows the pose estimation results during robot manipulation.

Table 4: Ablation Study.

Dataset	TYOL Dataset [114]				Annotated Real-World Dataset		
Method	w/ align	w/o coarse align	w/o fine align	w/o align	w/ diversified finetuning	w/ naive finetuning	w/o finetuning
AR(↑)	55.7	54.2	32.9	0.0	52.4	11.4	12.6
CD(↓)	0.72	0.94	1.05	38.47	–	–	–

Table 5: Real-world evaluation.

Methods	Success Rate(↑)
SRT3D [94]	6.7
DeepAC [117]	16.7
Ours	73.3

4.4 Ablation Study

Module Criticality Analysis. We conducted ablation studies on our object alignment method. As shown in Table 4, the full method achieves effective alignment. Removing the coarse alignment stage (*w/o coarse align*) led to a 1.5-point drop in AR and a 0.22 increase in CD. In contrast, removing the fine alignment stage (*w/o fine align*) caused a much larger degradation: AR decreased by 22.8 points and CD increased by 0.33. These results confirm that both stages contribute to performance, with fine alignment playing a particularly critical role in achieving high-quality results.

Efficacy of Synthetic Data Fine-Tuning. As shown in Table 4, we evaluate the impact of our text-guided generative augmentation on model fine-tuning on an annotated real-world dataset(Appendix. E). "Native fine-tuning" uses only the initial generated model, while "diversified fine-tuning" leverages texture-variant models from our generative strategy (Sec. 3.5). The latter yields a dramatic improvement: Average Recall (AR) jumps from 12.6% (no fine-tuning) to 52.4%—far outperforming native fine-tuning (11.4%). This confirms that our synthetic data, by diversifying textures while preserving structural consistency, effectively bridges the domain gap between generated models and real-world images. Critically, this strategy requires only one reference image per novel object, eliminating the need for costly labeled datasets—making it uniquely practical for real-world deployment. More details can be seen in Appendix. C.

5 Conclusion

We introduced OnePoseViaGen, a novel approach for one-shot 6D pose estimation of unseen objects without requiring 3D models. Our method leverages single-view 3D generation and introduces a coarse-to-fine alignment strategy to simultaneously recover the object's metric scale and 6D pose using multi-view feature matching and refinement iteratively. To enhance robustness, we employ text-guided generative domain randomization to create diverse synthetic training data, effectively bridging the gap between generated models and real objects. Extensive evaluations on challenging benchmarks (YCBINEOAT, TOYL, LM-O) demonstrate state-of-the-art performance, significantly outperforming prior methods, especially under occlusion and varying illumination. Successful real-world robotic grasping experiments further validate the practical applicability of our system. OnePoseViaGen represents a significant step towards versatile, data-efficient pose estimation.

5.1 Limitations

Although OnePoseViaGen achieves promising results across various datasets and real-world robotic experiments, it still faces challenges in handling deformable or articulated objects. In such cases, changes in object shape can lead to inaccurate 6D pose estimation. Future work will focus on incorporating test-time training into the inference pipeline, enabling continuous refinement of object geometry and accurate pose estimation for deformable objects. This direction aims to fully leverage the flexibility and generalization power of generative models in 6D pose estimation tasks.

References

[1] Z. Chen, A. Walsman, M. Memmel, K. Mo, A. Fang, K. Vemuri, A. Wu, D. Fox, and A. Gupta. Urdformer: A pipeline for constructing articulated simulation environments from real-world images. *arXiv preprint arXiv:2405.11656*, 2024.

[2] K. Srinivasan, J. Collins, E. Heiden, I. Ng, J. Bohg, and A. Garg. Dexmots: Dexterous manipulation with differentiable simulation. 2024.

[3] S. K. Dwivedi, D. Antić, S. Tripathi, O. Taheri, C. Schmid, M. J. Black, and D. Tzionas. Interactvlm: 3d interaction reasoning from 2d foundational models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22605–22615, 2025.

[4] L. Zhong, Y. Zhang, H. Zhao, A. Chang, W. Xiang, S. Zhang, and L. Zhang. Seeing through the occluders: Robust monocular 6-dof object pose tracking via model-guided video object segmentation. *IEEE Robotics and Automation Letters*, 5(4):5159–5166, 2020.

[5] H. Xia, E. Su, M. Memmel, A. Jain, R. Yu, N. Mbiziwo-Tiapo, A. Farhadi, A. Gupta, S. Wang, and W.-C. Ma. Drawer: Digital reconstruction and articulation with environment realism. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21771–21782, 2025.

[6] J. Yang, B. Ivanovic, O. Litany, X. Weng, S. W. Kim, B. Li, T. Che, D. Xu, S. Fidler, M. Pavone, et al. Emernerf: Emergent spatial-temporal scene decomposition via self-supervision. *arXiv preprint arXiv:2311.02077*, 2023.

[7] S. Xue, S. Cheng, P. Kachana, and D. Xu. Neural field dynamics model for granular object piles manipulation. In *Conference on Robot Learning*, pages 2821–2837. PMLR, 2023.

[8] Z. Chen, R. A. Potamias, S. Chen, and C. Schmid. Hort: Monocular hand-held objects reconstruction with transformers. *arXiv preprint arXiv:2503.21313*, 2025.

[9] A. Rashid, S. Sharma, C. M. Kim, J. Kerr, L. Y. Chen, A. Kanazawa, and K. Goldberg. Language embedded radiance fields for zero-shot task-oriented grasping. In *7th Annual Conference on Robot Learning*, 2023.

[10] Q. Wang, C. Deng, T. G. W. Lum, Y. Chen, Y. Yang, J. Bohg, Y. Zhu, and L. Guibas. Neural attention field: Emerging point relevance in 3d scenes for one-shot dexterous grasping. *arXiv preprint arXiv:2410.23039*, 2024.

[11] J. Wu, T. Xue, J. J. Lim, Y. Tian, J. B. Tenenbaum, A. Torralba, and W. T. Freeman. Single image 3d interpreter network. In *European Conference on Computer Vision*, pages 365–382. Springer, 2016.

[12] J. Kulhanek, S. Peng, Z. Kukelova, M. Pollefeys, and T. Sattler. Wildgaussians: 3d gaussian splatting in the wild. *arXiv preprint arXiv:2407.08447*, 2024.

[13] H. Matsuki, R. Murai, P. H. Kelly, and A. J. Davison. Gaussian splatting slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18039–18048, 2024.

[14] F. Dellaert, D. M. Rosen, J. Wu, R. Mahony, and L. Carlone. Shonan rotation averaging: Global optimality by surfing so (p) n. In *European Conference on Computer Vision*, pages 292–308. Springer, 2020.

[15] Q.-Y. Zhou, J. Park, and V. Koltun. Fast global registration. In *European conference on computer vision*, pages 766–782. Springer, 2016.

[16] C. Choy, J. Park, and V. Koltun. Fully convolutional geometric features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8958–8966, 2019.

[17] A. Simeonov, Y. Du, Y.-C. Lin, A. R. Garcia, L. P. Kaelbling, T. Lozano-Pérez, and P. Agrawal. Se (3)-equivariant relational rearrangement with neural descriptor fields. In *Conference on Robot Learning*, pages 835–846. PMLR, 2023.

[18] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1802–1811, 2017.

[19] X. Chen, H. Zhao, G. Zhou, and Y.-Q. Zhang. Pq-transformer: Jointly parsing 3d objects and layouts from point clouds. *IEEE Robotics and Automation Letters*, 7(2):2519–2526, 2022.

[20] Y. Wang, Z. Wang, M. Nakura, P. Bhowal, C.-L. Kuo, Y.-T. Chen, Z. Erickson, and D. Held. Articubot: Learning universal articulated object manipulation policy via large scale simulation. *arXiv preprint arXiv:2503.03045*, 2025.

[21] A. Zeng, K.-T. Yu, S. Song, D. Suo, E. Walker, A. Rodriguez, and J. Xiao. Multi-view self-supervised deep learning for 6d pose estimation in the amazon picking challenge. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1386–1383, 2017. doi:10.1109/ICRA.2017.7989165.

[22] D. Kappler, F. Meier, J. Issac, J. Mainprice, C. G. Cifuentes, M. Wüthrich, V. Berenz, S. Schaal, N. Ratliff, and J. Bohg. Real-time perception meets reactive motion generation. *IEEE Robotics and Automation Letters*, 3(3):1864–1871, 2018.

[23] B. Wen, W. Lian, K. Bekris, and S. Schaal. CatGrasp: Learning category-level task-relevant grasping in clutter from simulation. In *International Conference on Robotics and Automation (ICRA)*, pages 6401–6408, 2022.

[24] V. Blukis, T. Lee, J. Tremblay, B. Wen, I. S. Kweon, K.-J. Yoon, D. Fox, and S. Birchfield. One-shot neural fields for 3D object understanding. 2023.

[25] C. Zhuang, S. Li, and H. Ding. Instance segmentation based 6d pose estimation of industrial objects using point clouds for robotic bin-picking. *Robotics and Computer-Integrated Manufacturing*, 82:102541, 2023.

[26] B. Eisner, Y. Yang, T. Davchev, M. Vecerik, J. Scholz, and D. Held. Deep se (3)-equivariant geometric reasoning for precise placement tasks. *arXiv preprint arXiv:2404.13478*, 2024.

[27] Y. Zhao, M. Bogdanovic, C. Luo, S. Tohme, K. Darvish, A. Aspuru-Guzik, F. Shkurti, and A. Garg. Anyplace: Learning generalized object placement for robot manipulation. *arXiv preprint arXiv:2502.04531*, 2025.

[28] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg. Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. *arXiv preprint arXiv:1703.09312*, 2017.

[29] W. Shen, G. Yang, A. Yu, J. Wong, L. P. Kaelbling, and P. Isola. Distilled feature fields enable few-shot language-guided manipulation. *arXiv preprint arXiv:2308.07931*, 2023.

[30] J. Kerr, C. M. Kim, M. Wu, B. Yi, Q. Wang, K. Goldberg, and A. Kanazawa. Robot see robot do: Imitating articulated object manipulation with monocular 4d reconstruction. *arXiv preprint arXiv:2409.18121*, 2024.

[31] V. Caggiano, S. Dasari, and V. Kumar. Myodex: Generalizable representations for dexterous physiological manipulation.

[32] T. Chen, J. Xu, and P. Agrawal. A system for general in-hand object re-orientation. In *Conference on Robot Learning*, pages 297–307. PMLR, 2022.

[33] B. Romero, H.-S. Fang, P. Agrawal, and E. Adelson. Eyesight hand: Design of a fully-actuated dexterous robot hand with integrated vision-based tactile sensors and compliant actuation. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1853–1860. IEEE, 2024.

[34] T. Chen, M. Tippur, S. Wu, V. Kumar, E. Adelson, and P. Agrawal. Visual dexterity: In-hand reorientation of novel and complex object shapes. *Science Robotics*, 8(84):eadc9244, 2023.

[35] J. Bohg, A. Morales, T. Asfour, and D. Kragic. Data-driven grasp synthesis—a survey. *IEEE Transactions on robotics*, 30(2):289–309, 2013.

[36] D. Driess, F. Xia, M. S. Sajjadi, C. Lynch, A. Chowdhery, A. Wahid, J. Tompson, Q. Vuong, T. Yu, W. Huang, et al. Palm-e: An embodied multimodal language model. 2023.

[37] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020.

[38] Z. Wei, Z. Xu, J. Guo, Y. Hou, C. Gao, Z. Cai, J. Luo, and L. Shao. D (r, o) grasp: A unified representation of robot and object interaction for cross-embodiment dexterous grasping. *arXiv preprint arXiv:2410.01702*, 2024.

[39] A. Murali, B. Sundaralingam, Y.-W. Chao, W. Yuan, J. Yamada, M. Carlson, F. Ramos, S. Birchfield, D. Fox, and C. Eppner. Graspgen: A diffusion-based framework for 6-dof grasping with on-generator training. *arXiv preprint arXiv:2507.13097*, 2025.

[40] P. R. Florence, L. Manuelli, and R. Tedrake. Dense object nets: Learning dense visual object descriptors by and for robotic manipulation. *arXiv preprint arXiv:1806.08756*, 2018.

[41] A. Billard and D. Kragic. Trends and challenges in robot manipulation. *Science*, 364(6446): eaat8414, 2019.

[42] D. Guo, F. Sun, H. Liu, T. Kong, B. Fang, and N. Xi. A hybrid deep architecture for robotic grasp detection. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 1609–1614. IEEE, 2017.

[43] Y. Deng, X. Guo, Y. Wei, K. Lu, B. Fang, D. Guo, H. Liu, and F. Sun. Deep reinforcement learning for robotic pushing and picking in cluttered environment. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 619–626. Ieee, 2019.

[44] S. Sajjan, M. Moore, M. Pan, G. Nagaraja, J. Lee, A. Zeng, and S. Song. Clear grasp: 3d shape estimation of transparent objects for manipulation. In *2020 IEEE international conference on robotics and automation (ICRA)*, pages 3634–3642. IEEE, 2020.

[45] K. Ding, B. Chen, R. Wu, Y. Li, Z. Zhang, H.-a. Gao, S. Li, G. Zhou, Y. Zhu, H. Dong, et al. Preafford: Universal affordance-based pre-grasping for diverse objects and environments. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7278–7285. IEEE, 2024.

[46] T. Gervet, S. Chintala, D. Batra, J. Malik, and D. S. Chaplot. Navigating to objects in the real world. *Science Robotics*, 8(79):eadf6991, 2023.

[47] C. Forster, M. Pizzoli, and D. Scaramuzza. Svo: Fast semi-direct monocular visual odometry. In *2014 IEEE international conference on robotics and automation (ICRA)*, pages 15–22. IEEE, 2014.

[48] E. Kaufmann, A. Loquercio, R. Ranftl, A. Dosovitskiy, V. Koltun, and D. Scaramuzza. Deep drone racing: Learning agile flight in dynamic environments. In *Conference on Robot Learning*, pages 133–145. PMLR, 2018.

[49] M. Sabbah, K. Wojciechowski, H. Soh, D. Hsu, L. Righetti, N. Mansard, B. Watier, and V. Bonnet. Optimal motion prediction for human-to-robot handovers. 2025.

[50] M. Runz, M. Buffier, and L. Agapito. MaskFusion: Real-time recognition, tracking and reconstruction of multiple moving objects. 2018.

[51] E. Marchand, H. Uchiyama, and F. Spindler. Pose estimation for augmented reality: A hands-on survey. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 22(12): 2633–2651, 2015.

[52] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[53] O. M. Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, T. Kreiman, C. Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

[54] S. Venkataraman, Y. Wang, Z. Wang, Z. Erickson, and D. Held. Real-world offline reinforcement learning from vision language model feedback. *arXiv preprint arXiv:2411.05273*, 2024.

[55] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

[56] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pages 2165–2183. PMLR, 2023.

[57] Y. J. Ma, W. Liang, G. Wang, D.-A. Huang, O. Bastani, D. Jayaraman, Y. Zhu, L. Fan, and A. Anandkumar. Eureka: Human-level reward design via coding large language models. *arXiv preprint arXiv:2310.12931*, 2023.

[58] A. O'Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE, 2024.

[59] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

[60] V. Kumar, R. Shah, G. Zhou, V. Moens, V. Caggiano, A. Gupta, and A. Rajeswaran. Robohive: A unified framework for robot learning. *Advances in Neural Information Processing Systems*, 36:44323–44340, 2023.

[61] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023.

[62] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab. SSD-6D: Making RGB-Based 3D Detection and 6D Pose Estimation Great Again . In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1530–1538, Los Alamitos, CA, USA, Oct. 2017. IEEE Computer Society. doi:10.1109/ICCV.2017.169. URL https://doi.ieeecomputersociety.org/10.1109/ICCV.2017.169.

[63] S. Peng, Y. Liu, Q. Huang, X. Zhou, and H. Bao. Pvnet: Pixel-wise voting network for 6dof pose estimation. In *CVPR*, 2019.

[64] B. Wen, C. Mitash, S. Soorian, A. Kimmel, A. Sintov, and K. E. Bekris. Robust, occlusion-aware pose estimation for objects grasped by adaptive hands. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6210–6217, 2020. doi:10.1109/ICRA40945.2020.9197350.

[65] J. J. Lim, H. Pirsiavash, and A. Torralba. Parsing ikea objects: Fine pose estimation. In *Proceedings of the IEEE international conference on computer vision*, pages 2992–2999, 2013.

[66] C. Wang, D. Xu, Y. Zhu, R. Martín-Martín, C. Lu, L. Fei-Fei, and S. Savarese. Densefusion: 6d object pose estimation by iterative dense fusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3343–3352, 2019.

[67] C. Chi and S. Song. Garmentnets: Category-level pose estimation for garments via canonical space shape completion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3324–3333, October 2021.

[68] X. Li, H. Wang, L. Yi, L. J. Guibas, A. L. Abbott, and S. Song. Category-level articulated object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[69] W. Goodwin, S. Vaze, I. Havoutis, and I. Posner. Zero-shot category-level object pose estimation. Number 13699 in Lecture Notes in Computer Science, pages 516–532. Springer, 2022.

[70] G. Li, D. Zhu, G. Zhang, W. Shi, T. Zhang, X. Zhang, and J. Li. Sd-pose: structural discrepancy aware category-level 6d object pose estimation. In *Proceedings of the IEEE/CVF Winter Conference on applications of computer vision*, pages 5685–5694, 2023.

[71] J. Lin, Z. Wei, Z. Li, S. Xu, K. Jia, and Y. Li. Dualposenet: Category-level 6d object pose and size estimation using dual pose network with refined learning of pose consistency. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3560–3569, 2021.

[72] V. N. Nguyen, T. Groueix, M. Salzmann, and V. Lepetit. Gigapose: Fast and robust novel object pose estimation via one correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9903–9913, 2024.

[73] E. P. Örnek, Y. Labbé, B. Tekin, L. Ma, C. Keskin, C. Forster, and T. Hodan. Foundpose: Unseen object pose estimation with foundation features. In *European Conference on Computer Vision*, pages 163–182. Springer, 2024.

[74] B. Wen, W. Yang, J. Kautz, and S. Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17868–17879, 2024.

[75] Y. He, Y. Wang, H. Fan, J. Sun, and Q. Chen. Fs6d: Few-shot 6d pose estimation of novel objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6814–6824, 2022.

[76] Y. Labbé, L. Manuelli, A. Mousavian, S. Tyree, S. Birchfield, J. Tremblay, J. Carpentier, M. Aubry, D. Fox, and J. Sivic. MegaPose: 6D Pose Estimation of Novel Objects via Render & Compare. In *CoRL*.

[77] S. Moon, H. Son, D. Hur, and S. Kim. Co-op: Correspondence-based novel object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[78] J. Sun, Z. Shen, Y. Wang, H. Bao, and X. Zhou. Loftr: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8922–8931, 2021.

[79] J. Corsetti, D. Boscaini, C. Oh, A. Cavallaro, and F. Poiesi. Open-vocabulary object 6d pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18071–18080, 2024.

[80] X. Liu, G. Wang, R. Zhang, C. Zhang, F. Tombari, and X. Ji. Unopose: Unseen object pose estimation with an unposed rgb-d reference image. *arXiv preprint arXiv:2411.16106*, 2024.

[81] Y. Jin, V. Prasad, S. Jauhri, M. Franzius, and G. Chalvatzaki. 6dope-gs: Online 6d object pose estimation using gaussian splatting. *arXiv preprint arXiv:2412.01543*, 2024.

[82] C. Ye, Y. Wu, Z. Lu, J. Chang, X. Guo, J. Zhou, H. Zhao, and X. Han. Hi3dgen: High-fidelity 3d geometry generation from images via normal bridging. *arXiv preprint arXiv:2503.22236*, 3, 2025.

[83] M. Guo, M. Tang, H. Cha, R. Zhang, C. K. Liu, and J. Wu. Craft: Designing creative and functional 3d objects. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 7215–7224. IEEE, 2025.

[84] J. Xiang, Z. Lv, S. Xu, Y. Deng, R. Wang, B. Zhang, D. Chen, X. Tong, and J. Yang. Structured 3d latents for scalable and versatile 3d generation. *arXiv preprint arXiv:2412.01506*, 2024.

[85] J. J. Lim, H. Pirsiavash, and A. Torralba. Parsing ikea objects: Fine pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2013.

[86] H. Wang, S. Sridhar, J. Huang, J. Valentin, S. Song, and L. J. Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[87] Y. Liu, Y. Wen, S. Peng, C. Lin, X. Long, T. Komura, and W. Wang. Gen6d: Generalizable model-free 6-dof object pose estimation from rgb images. In *European Conference on Computer Vision*, pages 298–315. Springer, 2022.

[88] J. Sun, Z. Wang, S. Zhang, X. He, H. Zhao, G. Zhang, and X. Zhou. Onepose: One-shot object pose estimation without cad models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6825–6834, 2022.

[89] D. Huang, H. Ahn, S. Li, Y. Hu, and D. Lee. Estimation of 6d pose of objects based on a variant adversarial autoencoder. *Neural Processing Letters*, 55(7):9581–9596, Dec 2023. ISSN 1573-773X. doi:10.1007/s11063-023-11215-2. URL https://doi.org/10.1007/s11063-023-11215-2.

[90] C. Zhong, Y. Zheng, Y. Zheng, H. Zhao, L. Yi, X. Mu, L. Wang, P. Li, G. Zhou, C. Yang, et al. 3d implicit transporter for temporally consistent keypoint discovery. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3869–3880, 2023.

[91] C. Zhong, P. You, X. Chen, H. Zhao, F. Sun, G. Zhou, X. Mu, C. Gan, and W. Huang. Snake: Shape-aware neural 3d keypoint field. *Advances in Neural Information Processing Systems*, 35:7052–7064, 2022.

[92] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[93] X. He, J. Sun, Y. Wang, D. Huang, H. Bao, and X. Zhou. Onepose++: Keypoint-free one-shot object pose estimation without cad models. *Advances in Neural Information Processing Systems*, 35:35103–35115, 2022.

[94] M. Stoiber, M. Pfanne, K. H. Strobl, R. Triebel, and A. Albu-Schäffer. Srt3d: A sparse region-based 3d object tracking approach for the real world. *International Journal of Computer Vision*, 130(4):1008–1030, 2022.

[95] J. Xiang, Z. Lv, S. Xu, Y. Deng, R. Wang, B. Zhang, D. Chen, X. Tong, and J. Yang. Structured 3d latents for scalable and versatile 3d generation. *arXiv preprint arXiv:2412.01506*, 2024.

[96] T. Chen, C. Ding, S. Zhang, C. Yu, Y. Zang, Z. Li, S. Peng, and L. Sun. Rapid 3d model generation with intuitive 3d input. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12554–12564, 2024.

[97] Y. Chen, Y. Pan, H. Yang, T. Yao, and T. Mei. Vp3d: Unleashing 2d visual prompt for text-to-3d generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4896–4905, 2024.

[98] Y. Chen, C. Zhang, X. Yang, Z. Cai, G. Yu, L. Yang, and G. Lin. It3d: Improved text-to-3d generation with explicit view synthesis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 1237–1244, 2024.

[99] Z. Tang, J. Zhang, X. Cheng, W. Yu, C. Feng, Y. Pang, B. Lin, and L. Yuan. Cycle3d: High-quality and consistent image-to-3d generation via generation-reconstruction cycle. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 7320–7328, 2025.

[100] Z. Xie, Y. Zhang, X. Tang, Y. Wu, D. Chen, G. Li, and X. Jin. Styletex: Style image-guided texture generation for 3d models. *ACM Transactions on Graphics (TOG)*, 43(6):1–14, 2024.

[101] Z. Wu, P. Zhou, X. Yi, X. Yuan, and H. Zhang. Consistent3d: Towards consistent high-fidelity text-to-3d generation with deterministic sampling prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9892–9902, 2024.

[102] R. Xu, R. Chen, C. Tu, X. Gong, Z. Liu, L. Mei, X. Ren, and Z. Li. 3d models of sarcomas: the next-generation tool for personalized medicine. *Phenomics*, 4(2):171–186, 2024.

[103] T. Müller, A. Evans, C. Schied, and A. Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, 2022.

[104] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang. NeuS: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[105] B. Wen, J. Tremblay, V. Blukis, S. Tyree, T. Müller, A. Evans, D. Fox, J. Kautz, and S. Birchfield. BundleSDF: Neural 6-DoF tracking and 3D reconstruction of unknown objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 606–617, 2023.

[106] D. DeTone, T. Malisiewicz, and A. Rabinovich. SuperPoint: Self-Supervised Interest Point Detection and Description . In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 337–33712, Los Alamitos, CA, USA, June 2018. IEEE Computer Society. doi:10.1109/CVPRW.2018.00060. URL https://doi.ieeecomputersociety.org/10.1109/CVPRW.2018.00060.

[107] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich. Superglue: Learning feature matching with graph neural networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4937–4946, 2020.

[108] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. In *Readings in Computer Vision*, pages 726–740. Morgan Kaufmann, 1987.

[109] F. Poiesi and D. Boscaini. Learning general and distinctive 3d local deep descriptors for point cloud registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3): 3979–3985, 2022.

[110] T. Lee, B. Wen, M. Kang, G. Kang, I. S. Kweon, and K.-J. Yoon. Any6d: Model-free 6d pose estimation of novel objects. *arXiv preprint arXiv:2503.18673*, 2025.

[111] D. G. Lowe. Object recognition from local scale-invariant features. 1999.

[112] C. Gümeli, A. Dai, and M. Nießner. Objectmatch: Robust registration using canonical object correspondences. 2023.

[113] B. Wen, C. Mitash, and K. Bekris. Data-driven 6d pose tracking by calibrating image residuals in synthetic domains. *arXiv preprint arXiv:2105.14391*, 2021.

[114] T. Hodan, F. Michel, E. Brachmann, W. Kehl, A. GlentBuch, D. Kraft, B. Drost, J. Vidal, S. Ihrke, X. Zabulis, et al. Bop: Benchmark for 6d object pose estimation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 19–34, 2018.

[115] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother. Learning 6d object pose estimation using 3d object coordinates. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part II 13*, pages 536–551. Springer, 2014.

[116] Y.-W. Chao, W. Yang, Y. Xiang, P. Molchanov, A. Handa, J. Tremblay, Y. S. Narang, K. Van Wyk, U. Iqbal, S. Birchfield, et al. Dexycb: A benchmark for capturing hand grasping of objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9044–9053, 2021.

[117] L. Wang, S. Yan, J. Zhen, Y. Liu, M. Zhang, G. Zhang, and X. Zhou. Deep active contours for real-time 6-dof object tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14034–14044, 2023.

[118] M. Denninger, D. Winkelbauer, M. Sundermeyer, W. Boerdijk, M. Knauer, K. H. Strobl, M. Humt, and R. Triebel. Blenderproc2: A procedural pipeline for photorealistic rendering. *Journal of Open Source Software*, 8(82):4901, 2023. doi:10.21105/joss.04901. URL https://doi.org/10.21105/joss.04901.

[119] J. Wang and E. Olson. Apriltag 2: Efficient and robust fiducial detection. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4193–4198, 2016. doi:10.1109/IROS.2016.7759617.

[120] J. L. Schönberger and J.-M. Frahm. Structure-from-motion revisited. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, 2016. doi:10.1109/CVPR.2016.445.

[121] X. Contributors. Openxrlab structure-from-motion toolbox and benchmark. https://github.com/openxrlab/xrsfm, 2022.

[122] Z. Yu, T. Sattler, and A. Geiger. Gaussian opacity fields: Efficient adaptive surface reconstruction in unbounded scenes. *ACM Transactions on Graphics*, 2024.

[123] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick. Segment anything. *arXiv:2304.02643*, 2023.

Appendix

This appendix is organized as follows. In section A, we describe how carefully crafted text prompts guide the generation of diverse 3D models using Trellis [84], emphasizing the impact of linguistic variation on visual diversity. Section B details the construction of our synthetic training dataset, including the scene setup and statistical distributions across object pose, visibility, and distance. In section C, we report additional experimental results, including per-dataset performance analysis, failure mode characterization, and discussions of robustness across challenging conditions. In section D introduces a complete pipeline for constructing a ground-truth dataset for unseen objects in real-world scenes, encompassing scene acquisition, 3D reconstruction, coordinate alignment, and mask generation. Together, these components provide comprehensive support for evaluating our method in both synthetic and real-world conditions.

A Text Prompts and Model Output Diversity

As described in Section 3.5, we employ Trellis [84] to generate diverse textures. To this end, we use a prompt to instruct the VLM to produce suitable input prompts for Trellis:

Prompt: Given the image of a [OBJECT], generate a detailed and realistic prompt for a 3D modeling system to create diverse variants of this object. The prompt should request: A series of unique but plausible 3D models. Variations in design style (e.g., minimalist, industrial, futuristic, ergonomic). Inclusion of all essential functional components visible in the image. Use of realistic materials (e.g., matte plastic, brushed metal) with subtle imperfections for authenticity. Distinct color schemes with aesthetic and functional considerations (e.g., color-coded controls). Practicality and usability in real-world scenarios. Format the output as a natural-language instruction starting with "Generate...".

We then input the models generated based on an anchor image into our pipeline, resulting in the models shown in Fig. 5. It is evident that the generated models exhibit a rich diversity of textures, which helps further narrow the domain gap between the training dataset and real-world objects in the subsequent step of dataset generation. The experimental results validate the effectiveness of this approach, as discussed in the main text.

Figure 5: **Diversified Models.** This figure showcases the results of two types of diversified texture generation. For each object, the original model is displayed on the right, while the model with diversified textures is shown on the left. It can be observed that our proposed method, which utilizes text prompts to promote texture diversity, is capable of generating rich and varied surface textures. This contributes to the enhancement and diversification of training dataset generation.

B Statistical Distribution of the Training Dataset

The method for constructing a training dataset. First, we apply the method described in Sec. 3.5 to perform diversified texture generation on existing object models using text prompts, thereby creating 100 differently textured models as shown in Fig. 5, These models are then fed into the

Figure 6: **Generated training dataset.** This figure illustrates the dataset generated using our diversified texture models. It is evident that our generated dataset encompasses a rich variety of backgrounds, object poses, occlusion relationships, and lighting conditions. This contributes to narrowing the domain gap between the training dataset and real-world scenarios.

BlenderProc [118] rendering pipeline. For each synthetic scene, an initial model and three randomly selected diversified texture models are chosen as training targets; simultaneously, ten objects are randomly picked from the BOP dataset [114] to act as occluders, and a background environment is constructed by selecting a random texture map from the CCTextures. Subsequently, 100 camera positions are randomly determined with each camera oriented towards the geometric center of these 14 objects (4 targets + 10 occluders), incorporating eccentric noise into the camera positions while also introducing random perturbations to the rotation angles around their Z-axis to simulate variations in real-world shooting conditions. Through this process, we have constructed a large-scale, highly varied synthetic training dataset. Fig. 6 provides examples from this dataset, illustrating its rich variety of lighting conditions, degrees of occlusion, and object scales (distances from the camera), significantly enhancing the robustness and accuracy of the model in pose estimation tasks, as discussed in the main text.

The object pose distribution of the training dataset. We present the distributions of Azimuth, Elevation, Object Distance, and Visibility in the final generated dataset, as shown in Fig. 7. From these distributions, it is evident that our dataset exhibits a high degree of diversity in both azimuth and elevation angles, covering a wide range of horizontal and vertical orientations of objects relative to the camera. In terms of object distance, the dataset includes samples ranging from close-up to far-field scenarios, simulating the variation in depth at which objects may appear in real-world settings. Regarding visibility, due to the inclusion of occluders and randomized scene layouts, the extent of visible object regions varies significantly across images, resulting in diverse levels of occlusion. Taken together, the multi-dimensional diversity in our dataset closely mimics the complexity of real-world camera-captured scenes. This design choice effectively enhances the generalization capability and robustness of model training.

C Fine-tuning Details

When comparing with other one-shot methods, we did not fine-tune our model to ensure a fair comparison, making the evaluation valid under the same settings. In cases of object with long tail distribution where fine-tuning is desired, generating a diversified models takes around 10 minutes on an A800 GPU, while generating the training dataset requires approximately 10 minutes on an L20 GPU. As shown in Fig. 8, the fine-tuning stage, with training a single LoRA module, takes about

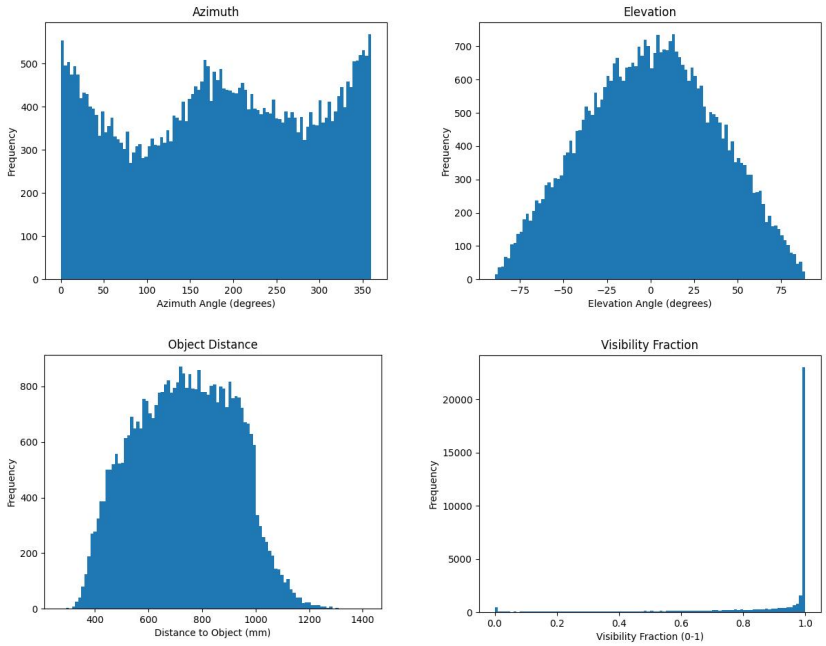

Figure 7: **Distribution Analysis of Training Dataset.** Top-left: Distribution of Azimuth angles for objects within the generated dataset; Top-right: Distribution of Elevation angles for objects; Bottom-left: Distribution of object distances from the camera; Bottom-right: Distribution of the proportion of object area visibility; These plots demonstrate that the constructed training dataset encompasses a rich diversity of poses and occlusion scenarios, underpinning its comprehensive variability and realism.

51 minutes to achieve 90% of highest AR on an single L20 GPU with 48GB of memory. Compared to redesigning the method from scratch, this fine-tuning process is computationally efficient and the time cost is acceptable in practice.

The detailed information of all modules in One-2-3-Pose is summarized in Table 6. Specifically, the Hi3DGen module is fine-tuned by concatenating the texture features with the original feature vectors. The Pose Generation module of FoundationPose is fine-tuned using the domain randomization strategy described in Section 3.5.

Table 6: Overview of modules and their properties.

Module Name	Input	Output	Is Fine-tuned?
Hi3DGen	Segmented anchor image	Normalized 3D mesh	✓
SuperPoint	RGB templates and segmented anchor image	Detected 2D keypoints with descriptors	✗
SuperGlue	Extracted 2D keypoints and descriptors	Matched 2D-2D correspondences	✗
PnP	Matched 2D-3D point correspondences and camera intrinsics	Estimated object pose (rotation R and translation t)	✗
Scale Optimization	3D model points in object coordinates and corresponding camera-space points	Scale factor s	✗
FoundationPose	Anchor image with segmentation mask and template under coarse pose	Refined object pose	✓
Trellis	Normalized 3D mesh and text-based prompt for texture generation	Texture-diversified 3D model	✗

D Additional experiments

Metrics. We assess performance using the metrics defined by the BOP challenge, specifically focusing on Average Recall (AR) for Visual Surface Discrepancy (VSD), Maximum Symmetry-aware Surface Distance (MSSD), and Maximum Symmetry-aware Projection Distance (MSPD) [114]. These metrics offer complementary insights into pose accuracy by evaluating recall rates across

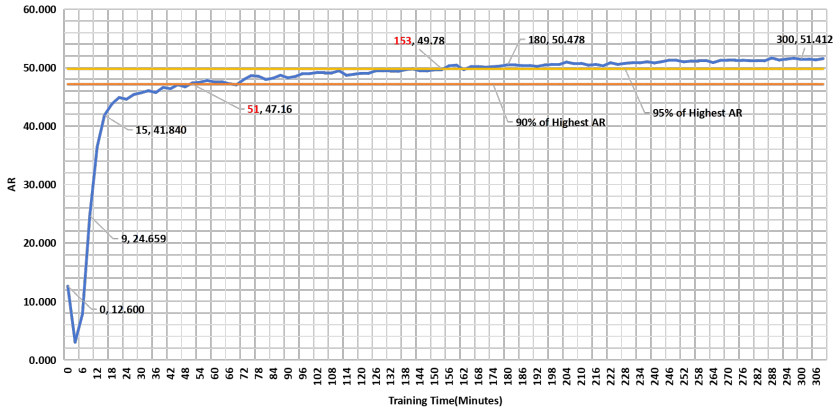

Figure 8: **Finetune Performance with Time.**

Table 7: Processing Stages and Time Consumption

Stage	Submodule	Time Cost (s)
Model Generation	Model Generation	9.10
	Model Export	12.84
Scale Recovery (1 Round)	Coarse Alignment	0.75
	Fine Alignment	0.23
Pose Estimation (FoundationPose)	Pose Generation	0.88
	Pose Selection	0.42

various thresholds. This comprehensive evaluation approach ensures a thorough assessment of algorithmic performance in diverse scenarios, reflecting different aspects of precision and robustness in pose estimation.

Performance Analysis on the LM-O Dataset. The LM-O dataset [115] comprises 12 objects, predominantly distinguished by their lack of texture and frequent occlusion. Performance metrics for each object are summarized in Table 8, where τ denotes the misalignment tolerance. As shown in Fig. 9, performance degrades when the target object occupies a small region in the image. Nonetheless, the method still achieves relatively robust results under such challenging conditions. As observed in Table 8, the model exhibits stable pose estimation across most categories when the translational error threshold exceeds 0.15, indicating consistent precision and reliability in 6D pose estimation. However, the "ape" category shows notably lower accuracy, primarily due to its minimal textural information and ambiguous geometry. These characteristics result in a mismatch between the reconstructed and ground-truth models, which in turn hampers accurate model alignment and pose estimation. Despite this limitation, as illustrated in Fig. 11, the proposed method demonstrates strong and consistent performance across the majority of object categories, highlighting its generalization capability under diverse shape and textural conditions in real-world 6D object pose estimation tasks.

Table 8: Detailed Metrics on the LM-O Dataset.

Metrics	object								avg
	ape	can	cat	driller	duck	eggbox	glue	holepunch	
MSPD	79.6	93.7	80.8	82.1	84.4	86.0	77.9	91.5	84.9
MSSD	64.8	92.6	71.2	85.8	72.9	80.1	75.8	87.9	79.6
VSD	37.5	75.5	57.7	71.2	66.1	56.6	51.9	58.2	60.0
AR	60.6	87.3	69.9	79.7	74.5	74.2	68.5	79.2	74.8

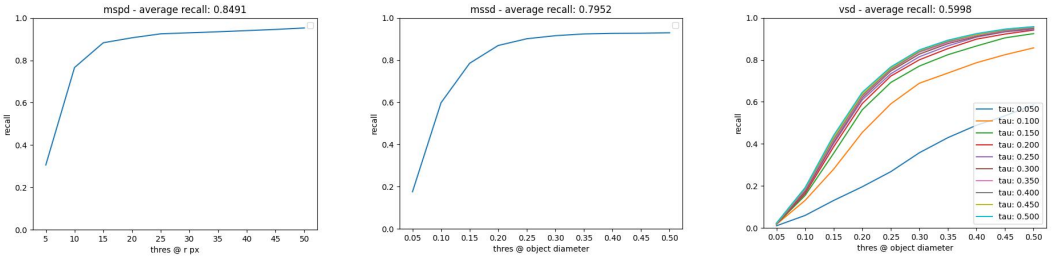

Figure 9: **Visualization of Metrics on the LM-O Dataset.** Left: Illustration of the MSPD metric variation with respect to the visible size of objects in pixels. Middle: Demonstration of the MSSD metric variation according to the object size in meters. Right: Presentation of the VSD metric variation with respect to τ (denoting the misalignment tolerance) and object size. These graphs effectively capture the metrics' dependencies on object visibility, size, and alignment tolerance, providing insights into their varying influences under different conditions.

Figure 10: **Performance in LM-O dataset.** In each image, the red, green, and blue lines represent the x, y, and z axes of the model, respectively, while the pink line shows the rendered contour under the estimated pose. By comparing the rendered contour with the ground-truth outline of the object, it is evident that our method is highly robust, performing well across various objects and under different occlusion scenarios.

Performance Analysis on the TOYL Dataset. The TOYL [114] test dataset contains 21 objects, primarily distinguished by complex lighting conditions and the fact that most objects are positioned relatively far from the camera. Performance metrics for each object are summarized in Table 9. As illustrated in Fig. 12, performance declines when the target occupies only a small portion of the image, which is often due to the considerable distance from the camera. This makes texture the primary discriminative cue for pose estimation, as the objects' high symmetry reduces the effectiveness of other features. Despite these challenges, the model exhibits relatively stable performance across most object categories when the translational error threshold exceeds 0.15, indicating consistent behavior under varying conditions. As shown in Table 9, the method faces notable difficulties with objects numbered 04 and 18. Their high symmetry restricts reliable orientation estimation to surface texture cues alone, often leading to mismatches when the opposite side of the object is visible. Nevertheless, for the majority of other objects, the proposed method achieves stable and accurate pose estimation, demonstrating strong robustness and adaptability even when objects are positioned at significant distances from the camera.

Figure 11: **Comparison Between Original and Generated Models on LM-O dataset.** The first row displays the original object models, while the second row shows the generated models under the same pose. The third row presents a bottom-view comparison of the generated models. As can be seen, the generated models exhibit high quality and closely resemble the original objects in terms of texture and structure, demonstrating the effectiveness of our generation and scale recovery approach.

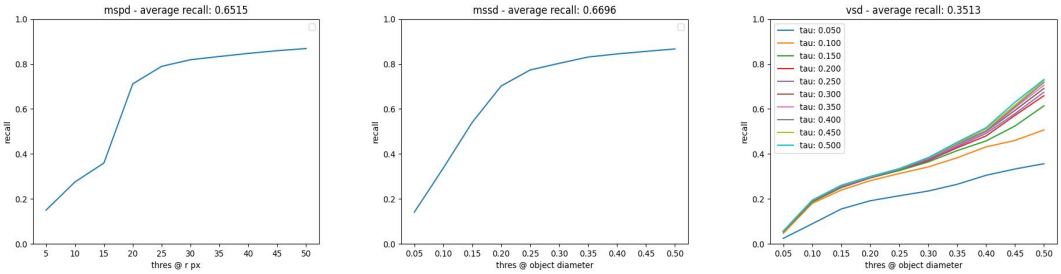

Figure 12: **Visualization of Metrics on the TYOL Dataset.** Left: Illustration of the MSPD metric variation with respect to the visible size of objects in pixels. Middle: Demonstration of the MSSD metric variation according to the object size in meters. Right: Presentation of the VSD metric variation with respect to τ (denoting the misalignment tolerance) and object size. These graphs effectively capture the metrics' dependencies on object visibility, size, and alignment tolerance, providing insights into their varying influences under different conditions.

Failure modes. As shown in Fig. 15 and Fig. 16, the proposed method exhibits performance limitations in scenarios involving severe occlusion, strong motion blur, as well as objects with low texture and high symmetry.

More details about the real-world experiments. The primary perception system for both tasks relied on Intel RealSense cameras, providing RGB and depth information at a resolution of 640×480 and a frame rate of 30 Hz. For **Task 1** (ROKAE Pick-and-Place), the camera was fixed next to the robotic arm, approximately 0.5 meter above the center of the manipulation area, angled at 15 degrees from the vertical. For **Task 2** (Dual-Arm AgileX PiPER Manipulation), an eye-in-hand camera on each arm, offers close-up views beneficial for fine manipulation, whereas a fixed external camera provides a broader view. All cameras were calibrated prior to the experiments using standard procedures to ensure accurate spatial information.

Task 1: Pick-and-Place with ROKAE Arm and XHAND1 This task evaluated the ability of a ROKAE collaborative robot arm equipped with an XHAND1 dexterous hand to pick up a variety of objects from a starting location and place them accurately at a designated target pose.

1. Hardware Configuration.

Table 9: Detailed Metrics on the TOYL Dataset.

Metrics	object																					avg
	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	
MSPD	80.1	48.5	50.5	36.8	72.1	46.9	74.2	42.2	85.1	82.2	77.8	84.6	83.1	68.9	80	72.5	59.5	31.9	74.4	65.3	54.5	55.7
MSSD	76.4	51.1	40.8	33.6	75.9	47.5	73	63.3	91.6	83.1	76.1	84.7	86.1	79.8	88.5	77.7	48.3	26	90.3	61.8	54.1	65.1
VSD	31.1	33.6	41.7	18.5	25.5	35	31.4	24.9	42.8	38.1	25.6	45.6	47.5	52.8	51.3	44.3	40.5	26.7	34.4	27.3	20.1	67.0
AR	62.5	44.4	44.3	29.6	57.8	43.1	59.5	43.5	73.2	67.8	59.8	71.6	72.2	67.2	73.3	64.8	49.4	28.2	66.4	51.5	42.9	35.2

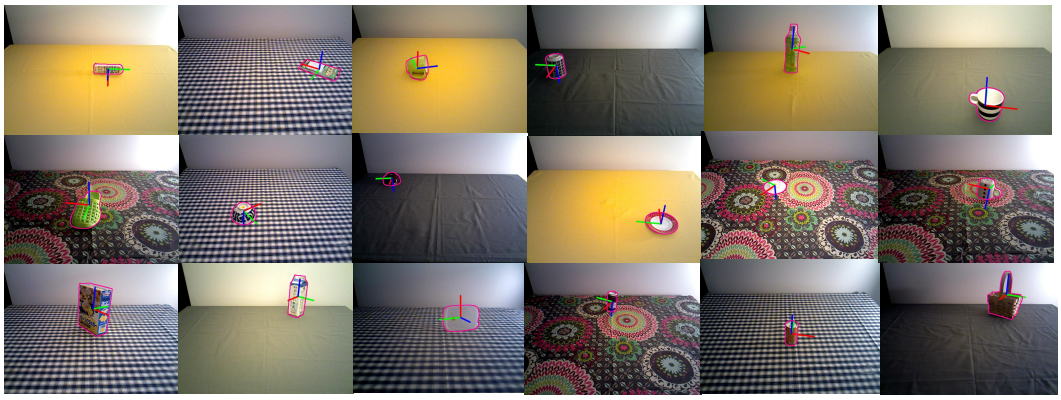

Figure 13: **Performance in TYOL dataset.** In each image, the red, green, and blue lines represent the x, y, and z axes of the model, respectively, while the pink line shows the rendered contour under the estimated pose. By comparing the rendered contour with the ground-truth outline of the object, it is evident that our method is highly robust, performing well across various objects and under different occlusion scenarios.

1. ROKAE Robot Arm ($\times 1$), is designed for collaborative applications, featuring a cabinet-free design and integrated torque sensors in each joint, promoting deployment flexibility and safety, with 6-DoF.

2. XHAND1 Dexterous Hand ($\times 1$), has 12 active degrees of freedom and supports force control with haptic sensor feedback, allowing us to adapt to object shapes and ensure stable grasps. The hand is equipped with five 270° three-dimensional encircling tactile array sensors on the fingertips, providing a tactile resolution of 12×10 per fingertip and sensing 3D forces including tangential components (X and Y). Joint sensors provide position, velocity, temperature, and current (torque) information.

The XHAND1 was mechanically attached to the ROKAE arm's end-effector flange.

2. Task Protocol. A set of 15 diverse objects was used to evaluate the pick-and-place capabilities. Objects were presented randomized within a $10 \times 10cm^2$ area with orientation randomized ± 15 degrees around the vertical axis. The target placement pose for each object was a plate into which the object needed to be placed.

For the 12-DoF XHAND1, grasp planning involved a pre-defined grasp synergies. Tactile and force feedback from the XHAND1's sensors were used to confirm contact, and monitor grasp stability during lift and transport, leveraging its force-position control capabilities.

An initial 6D pose estimate of the object was obtained from the perception system (detailed in the main paper). This pose was used to determine the robot's approach trajectory to the object and plan the motion to the target placement location. The system operated on the assumption that this initial pose was sufficiently accurate for grasp execution. Then we use the pose tracking methods to guide the motion of the arm in real-time.

For each of the 15 objects, 30 pick-and-place trials were conducted. Between trials, the object was randomized within the defined start region, and the scene was reset.

3. Success Criteria.

Figure 14: **Comparison Between Original and Generated Models on TOYL dataset.** The first row displays the original object models, while the second row shows the generated models under the same pose. The third row presents a bottom-view comparison of the generated models. As can be seen, the generated models exhibit high quality and closely resemble the original objects in terms of texture and structure, demonstrating the effectiveness of our generation and scale recovery approach in TOYL dataset.

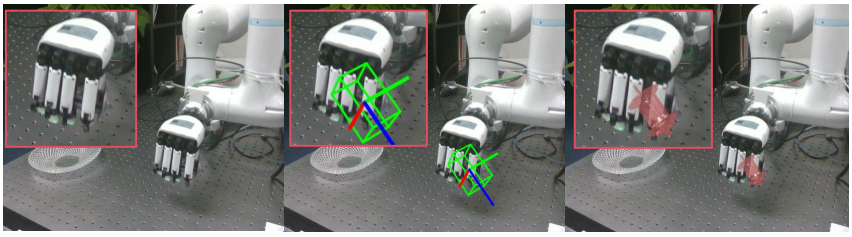

Figure 15: **Failure modes in real-world test.** Left: The original image. Middle: Detection results showing the bounding box and model's XYZ axes. Right: Rendered model image based on the detected pose. This visualization demonstrates that our method struggles in scenarios characterized by high symmetry, significant occlusion, and severe motion blur. Despite these challenges, the method shows promise in more favorable conditions, highlighting areas for potential improvement.

1. Successful Grasp: object was securely held by the XHAND1, lifted at least 5 cm clear of the support surface, and maintained a stable grasp without slipping or being dropped during the initial lift.

2. Successful Transport: The object was moved from the pick location to the designated target area without any collisions with the environment or being dropped.

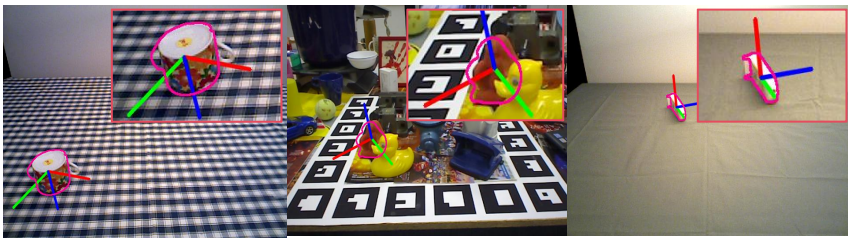

Figure 16: **Failure modes in LMO and TOYL dataset.** In each image, the red, green, and blue lines represent the x, y, and z axes of the model, respectively, while the pink line shows the rendered contour under the estimated pose. The top-right corner of each image displays an enlarged view of the corresponding failure region.

3. Successful Placement: The object was released at the target pose, the object had to remain stable in its placed configuration for at least 3 seconds after the XHAND1 retracted, without toppling or rolling away.
4. Overall Trial Success: A trial was deemed successful if and only if all three criteria (Grasp, Transport, and Placement) were met.

Task 2: Dual-Arm Manipulation (Pick, Handoff, Place) with AgileX PiPERs This task involved two AgileX PiPER robot arms collaboratively picking an object, handing it off from one arm to the other, and then placing it at a final target location. This sequence introduces challenges related to inter-arm coordination, synchronization, and stable object transfer.

1. Hardware Configuration.

1. AgileX PiPER Robot Arms ($\times 2$), the two PiPER arms were mounted on fixed bases, facing each other across a central workspace, 800 mm apart. Each PiPER arm was 6-DoF, equipped with an Agilex Pika Gripper.

2. Task Protocol. Similar to Task 1, objects were presented randomized within a $10 \times 10cm^2$ area with orientation randomized ± 15 degrees around the vertical axis. Arm 1 was set as Giver, while Arm 2 was Receiver.

A designated handoff procedure was followed. The giving arm (Arm 1) moved the object to a predefined handoff region (a specific area in the shared workspace). Arm 2 approached the object in Arm 1's gripper with a same grasp configuration, and then the receiving arm (Arm 2) placed the object at the table.

Arm 1 always performed the initial pick and acted as the "giver"; Arm 2 always acted as the "receiver" and performed the final place. Arm motions were synchronized using an event-based system where Arm 2's approach was triggered by Arm 1 reaching the handoff pose, and Arm 1's release was triggered by a confirmation signal from Arm 2 (successful grasp confirmed by force sensors). The timing of gripper actions (Arm 2 closing, Arm 1 opening) was critical to prevent drops.

An initial 6D pose estimate of the object was used by Arm 1 for the pick. During handoff, Arm 2 relied on the known pose of Arm 1's end-effector and visually track the object in Arm 1's gripper before Arm 2 grasped it. After handoff, the pose of the object was used by Arm 2 for planning the final placement.

For each of the 15 objects, 30 pick-handoff-place trials were conducted. Scene and object reset procedures were followed between trials.

3. Success Criteria.

1. Successful Pick (Arm 1): The object was securely grasped by Arm 1 and moved to the designated pre-handoff pose without slipping, dropping, or collision.
2. Successful Handoff: Arm 2 securely grasped the object from Arm 1, arm 1 released the object only after Arm 2's grasp was confirmed via threshold met on the gripper force sensor (5 N). The object was not dropped during the transfer from Arm 1 to Arm 2 and the object was stable in Arm 2's gripper after Arm 1 retracted.
3. Successful Place (Arm 2): The object was released by Arm 2 at the table, without slipping or being dropped.
4. Overall Task Success: A trial was considered successful if and only if all three stages (Pick, Handoff, and Place) were completed successfully.

E Pipeline for Constructing a Ground Truth Real-world Dataset

Scene data acquisition and preprocessing. In this paragraph, we describe the complete pipeline for scene data acquisition and preprocessing, which includes four key stages:

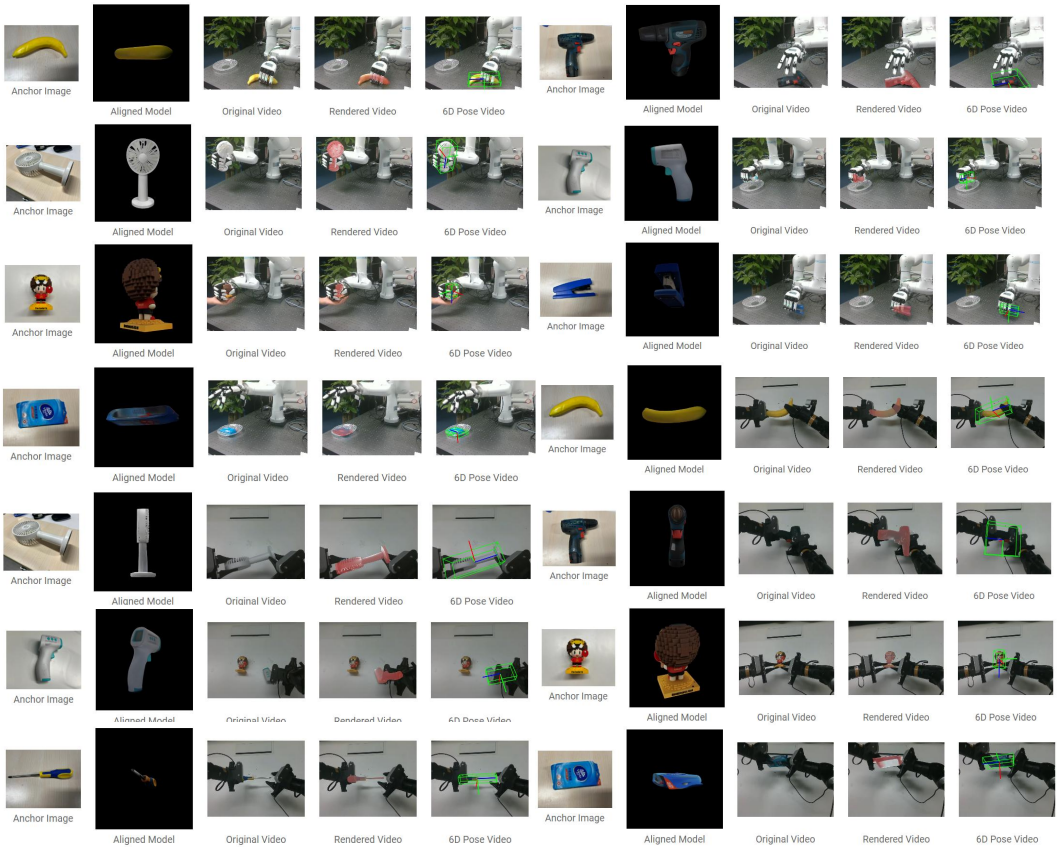

Figure 17: **Real-world Robot Manipulation Examples.** We recommend that readers visit the webpage to watch a dynamic demonstration video.

1. **Deployment of AprilTags and Video Recording:** To facilitate accurate camera pose calculation and metric scale recovery, multiple high-contrast and easily detectable AprilTags [119] are strategically placed throughout the target scene. As shown in Fig. 18,these tags serve as known geometric reference points, significantly improving the robustness and accuracy of the reconstruction process. A monocular camera is then used to record a video sequence at a fixed frame rate, ensuring full coverage of the most of AprilTags from multiple viewpoints.

2. **Feature Matching and Sparse Reconstruction Using COLMAP:** Structure-from-Motion (SfM) [120] processing is performed using COLMAP. First, local feature descriptors are extracted from each frame. These features are then matched across images, followed by geometric verification to remove outliers. Finally, an incremental SfM algorithm jointly optimizes both intrinsic and extrinsic camera parameters, resulting in a sparse 3D point cloud. This sparse reconstruction provides initial geometric priors for subsequent dense reconstruction.

3. **Distortion Correction:** Due to radial and tangential distortions inherent in standard camera lenses, which can degrade reconstruction quality, it is essential to perform distortion correction. The raw images are undistorted and mapped to a distortion-free coordinate system. Simultaneously, the corresponding camera parameter files are updated to reflect this transformation, providing more accurate input for the 3D reconstruction pipeline.

4. **Scale Estimation and Point Cloud Calibration Using XRSFM:** In purely vision-based SfM pipelines, the reconstructed scene is typically only defined up to a similarity transformation, lacking true metric scale. By leveraging the known physical dimensions of

the AprilTags, the XRSFM [121] method is employed to compute the global scaling factor. This allows the sparse point cloud to be transformed from a similarity space into a Euclidean space with real-world units, enabling downstream tasks that require metric accuracy.

3D Reconstruction and Object Mesh Extraction. After metric calibration of the point cloud, we utilize Gaussian Opacity Fields (GOF) [122] to perform neural implicit field modeling of the scene. This method embeds sparse point cloud data into a Gaussian representation and models surface geometry through a learnable opacity field. Specifically, the point cloud is first initialized, and a set of spatially supported Gaussian attributes is constructed. The model is then trained using radiance optimization combined with geometry-aware training strategies, enabling the implicit representation of fine-grained surface details. Finally, a triangle mesh is extracted from the learned GOF representation via marching cubes or analogous voxelization techniques, yielding a high-quality surface reconstruction of the scene. The target object mesh is then manually segmented from the reconstructed scene mesh for further processing.

Coordinate alignment and camera/object pose estimation. This paragraph introduces the complete workflow for coordinate alignment and 6D pose estimation, including initial camera pose initialization, camera poses calculation, and per-frame object pose computation:

1. **Aligning Object and Scene Meshes for Initial Camera Pose Estimation:** The object mesh is rigidly aligned with the scene mesh using ICP (Iterative Closest Point) registration algorithms. This provides the object's absolute position within the scene. Combined with AprilTag location information, this enables derivation of the initial camera pose $\mathbf{T}_{C_0}^{W}$ (rotation and translation matrices in the world coordinate system) for the first frame.
2. **Estimating Per-Frame Camera Poses Using COLMAP Features:** Based on the camera poses $\mathbf{T}_{C_i}^{W}$ output by COLMAP (relative to the sparse point cloud), and incorporating the previously estimated metric scale, all camera poses are transformed into the unified world coordinate system. This step establishes a reliable camera trajectory that supports accurate object position computation.
3. **Computing Object Poses Across Frames:** Assuming the object remains static in the scene, its pose in the first frame can be transformed into any other frame's camera coordinate system through a rigid transformation chain. Specifically, given the object's pose $\mathbf{T}_{O_1}^{W}$ in the world coordinate system (where subscript O_1 denotes the object in the first frame and superscript W denotes the world coordinate system), and the camera poses $\mathbf{T}_{C_i}^{W}$ for each frame i, we can compute the object's pose $\mathbf{T}_{O_i}^{C_i}$ in the camera coordinate system of frame i using the following transformation:

$$\mathbf{T}_{O_i}^{C_i} = (\mathbf{T}_{C_i}^{W})^{-1} \cdot \mathbf{T}_{O_1}^{W} \tag{2}$$

   Here, $\mathbf{T}_{C_i}^{W}$ represents the transformation matrix from the world coordinate system to the camera coordinate system of frame i, and $(\mathbf{T}_{C_i}^{W})^{-1}$ is its inverse, which transforms points from the camera coordinate system back to the world coordinate system.

   The 6-DoF object pose in each image frame consists of three rotational parameters ($\mathbf{R}_{O_i}^{C_i}$) and three translational parameters ($\mathbf{t}_{O_i}^{C_i}$). These can be extracted from the transformation matrix $\mathbf{T}_{O_i}^{C_i}$ as follows:

$$\mathbf{T}_{O_i}^{C_i} = \begin{bmatrix} \mathbf{R}_{O_i}^{C_i} & \mathbf{t}_{O_i}^{C_i} \\ 0 & 1 \end{bmatrix} \tag{3}$$

Object mask generation and visualization details. This paragraph introduces the methodology for generating accurate object masks and visualizing key details, which includes two main steps:

1. **Rendering Object Masks Using 3D Models and Camera Poses:** Given the known 3D mesh model of the object and the current frame's camera parameters (intrinsic + extrinsic), a binary object mask is rendered onto the image plane. This forward projection process generates a foreground-background segmentation that supports downstream tasks such as pose estimation and visibility analysis.
2. **Extracting Visible Region Masks Using Segmentation Methods (e.g., SAM 2):** To further improve mask accuracy—especially under occlusion, a pre-trained instance segmentation method such as SAM 2 [123] is applied to the input images. This generates a visible region mask highlighting only the observable parts of the object, which can be used during training or evaluation to exclude occluded or invisible regions and improve robustness.

Figure 18: **Testing dataset examples.** The dataset we have constructed is showcased from various views. It can be seen that our testing dataset includes a range of object poses, occlusion relationships, and distances from the camera, demonstrating its richness and diversity.

Finally, we constructed the following test dataset. The dataset visualization is shown in Fig. 18, and its distribution is illustrated in Fig. 19. These figures demonstrate that the dataset exhibits diverse object positions and viewing angles, making it well-suited for comprehensively evaluating the performance of the trained models.

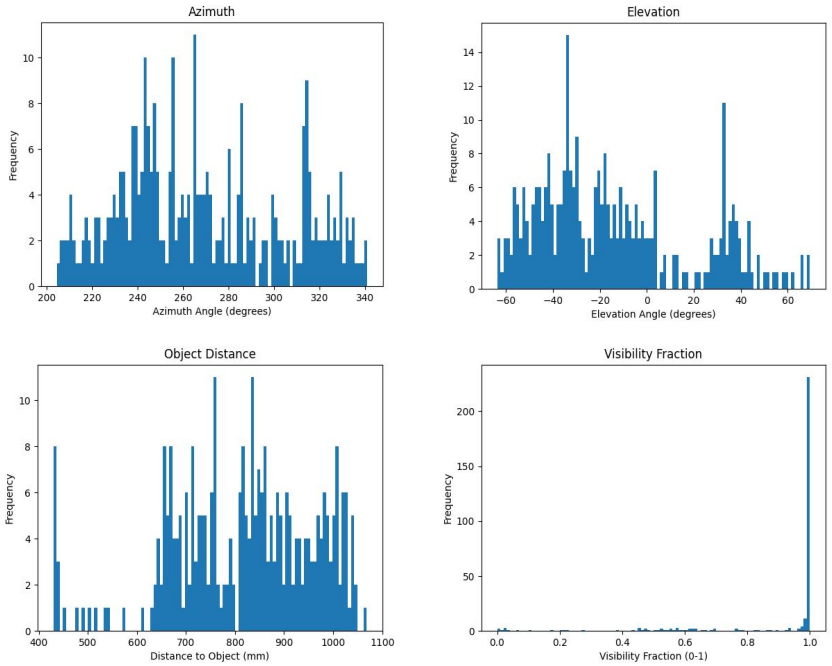

Figure 19: **Distribution Analysis of Testing Dataset.** Top-left: Distribution of Azimuth angles for objects within the generated dataset; Top-right: Distribution of Elevation angles for objects; Bottom-left: Distribution of object distances from the camera; Bottom-right: Distribution of the proportion of object area visibility; These plots demonstrate that the constructed testing dataset encompasses a rich diversity of poses and occlusion scenarios, underpinning its comprehensive variability and realism.